Mex # 5 Fasttext Embedding

For the Mex # 5, I implemented a skip-gram model for fasttext with CUDA implementation to train 750 documents. The implementation was based off from *Enriching Word Vectors with Subword Information* from Bojanowski, P., et. Al. The model collects positive and negative word-context pairs from a sentence and computes the dot products of word vectors and context vectors, incorporating n-gram representations to capture sub-word information. Using stochastic gradient descent (SGD), the word vectors and context vectors are updated in parallel on the GPU for both positive and negative samples by launching CUDA kernels. The updated vectors are then transferred back from GPU to CPU where the vocabularies are updated separately.

$$s(w,c) = \sum_{g \in G_w} z_g^T \, v_c$$

$$L = \; l\big(s(w,c)\big) + \sum_{n \in \mathcal{N}_{t,c}} l(-s(w,n_i))$$

$$\frac{\delta L}{\delta w} = \; l\big(s(w,c)\big) + \sum_{n \in \mathcal{N}_{t,c}} l(-s(w,n_i))$$

Where:

$l\colon\ x \ \mapsto \log\,(1+e^{-x})$ is the logistic function

$\mathcal{N}_{t,c}$ is a set of negative samples

$G_w \subset \{1,\dots,G\}$ the set of n-grams appearing on w.

For positive samples:

$$w \ \leftarrow w - \ \eta\big(l\big(s(w,c)\big) - 1\big)c$$

$$c \ \leftarrow c - \ \eta\big(l\big(s(w,c)\big) - 1\big)w$$

For negative samples:

$$w \leftarrow w - \eta\big(l\big(s(w, n_i)\big) - 1\big)c$$

Where:

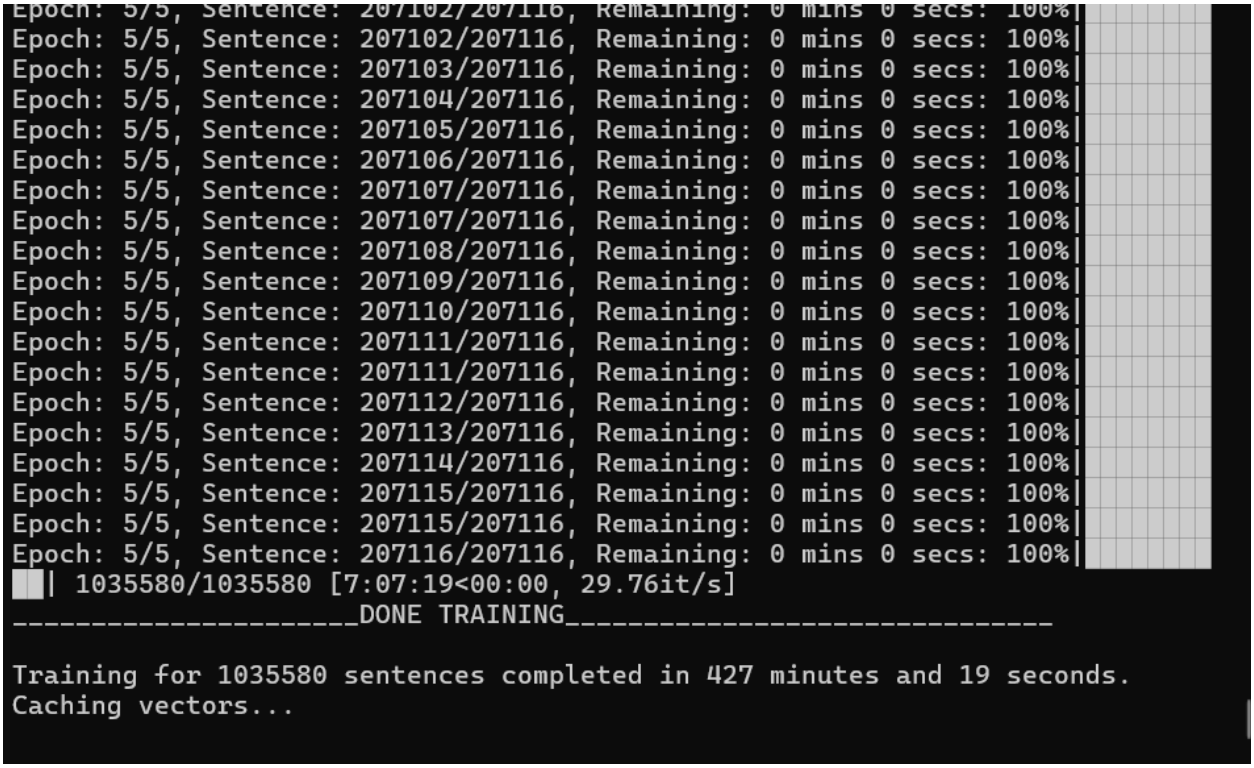
w is the word vector

c is the context vector

η is the gradient

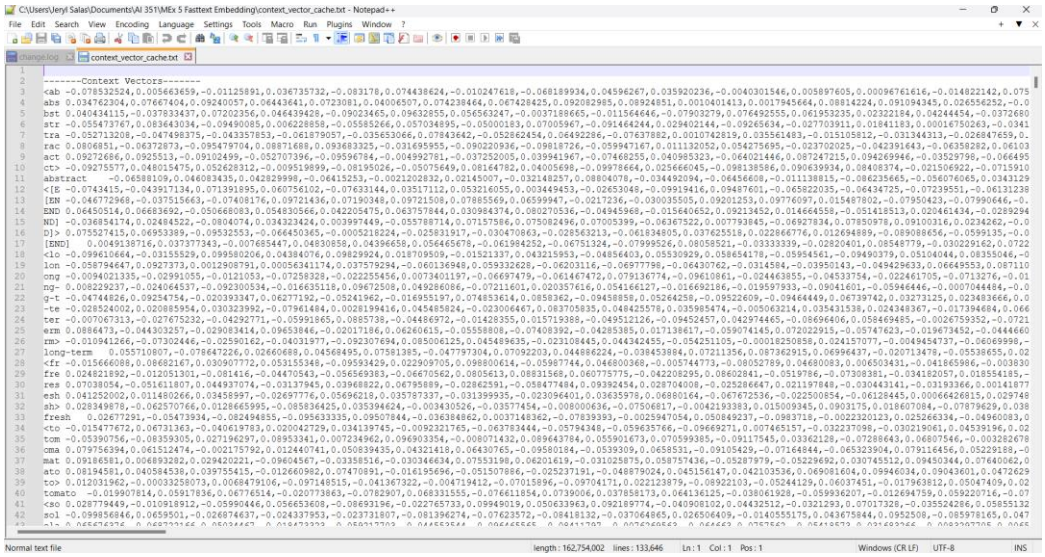
Results:

My custom model took 427 minutes and 19 seconds to train 1,035,580 sentences while the Gensim model only took 3 minutes and 2 seconds. I saw from my task manager, that around 50-60% of my GPU device (0) is used for my custom model while for the Gensim model, 100% of my CPU memory is used. What I'm getting at is maybe I haven't utilized all the threads that I have in the GPU unlike in Gensim where it was able to utilize most of my CPU memory. I was able to print the word, context and n gram vector embeddings on a .txt file as well as the comparison of the Gensim and my custom model on sample sentences. I was able to get a mean difference in the range of 0.2 to 0.3.

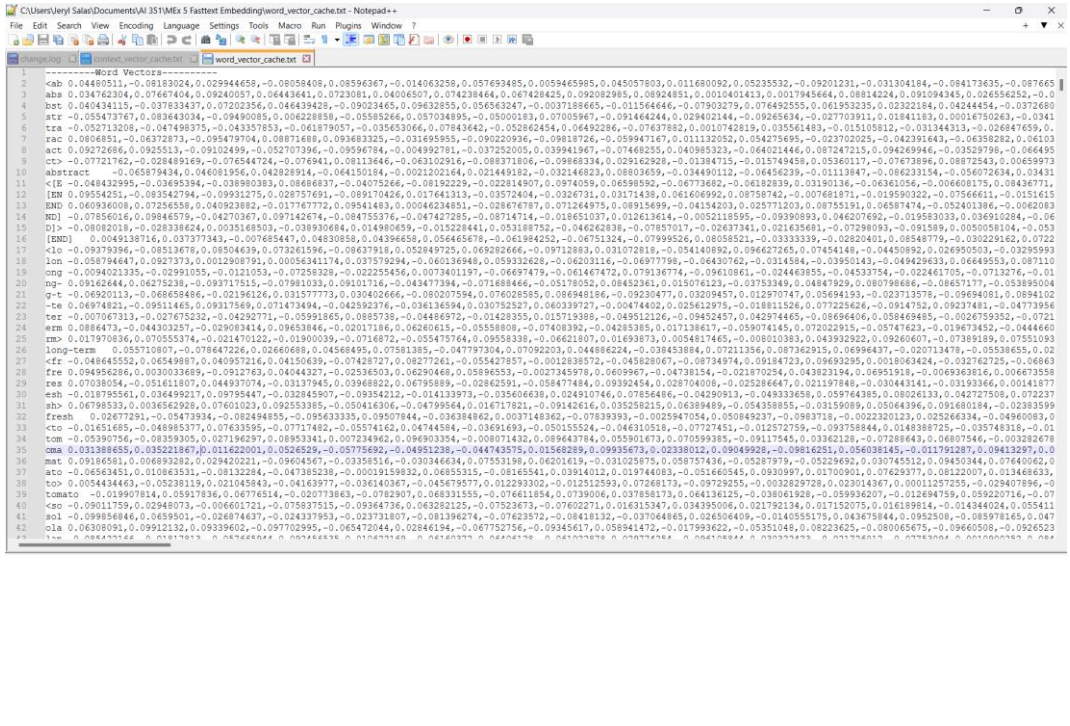


Screenshot of .txt files:

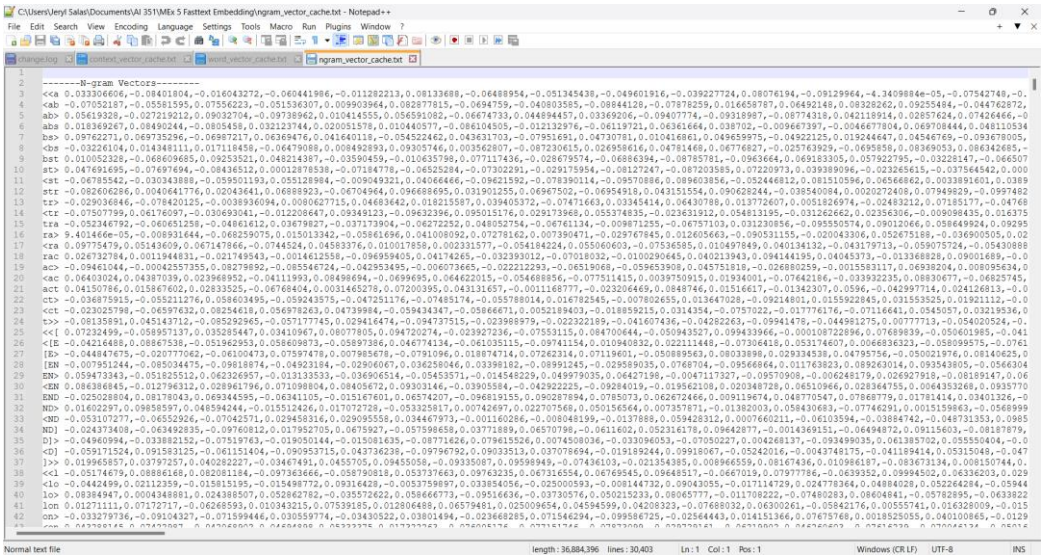
1.) Word vector



2.) Context vector



3.) N gram vector



4.) Comparison